How Data Travels Through the Internet

By Hasindu Ohasa

April 26, 2025

Table of Contents

Introduction

Every time you send an email, browse a website, or video-chat with a friend, data is racing across the globe in milliseconds. This happens thanks to the Internet's sophisticated architecture of protocols and hardware. In this paper, we will demystify **how data travels through the Internet**, step by step. We explain the key technologies—from packet switching and TCP/IP layering to routers and cables—and follow a typical packet journey. Along the way we cover concepts like error checking, congestion control, and encryption. By the end, you'll have a detailed picture of the Internet's data highway in friendly, accessible terms.

Packet Switching vs. Circuit Switching

Data networks can use **circuit switching** or **packet switching** to move information. In a **circuit-switched network**, a dedicated communication path is established between endpoints before any data is sent. The classic example is an old-style telephone call: you "dial" and a continuous path is reserved for the entire call (Packet-Switched Network vs. Circuit-Switched Network) (Packet-Switched Network vs. Circuit-Switched Network). All the data flows steadily over that circuit until one side hangs up. This guarantees a constant bandwidth and ordered delivery, but even when one end is silent, the channel stays reserved and cannot be used by others (Packet-Switched Network vs. Circuit-Switched Network) (Packet-Switched Network vs. Circuit-Switched Network). In contrast, the Internet uses **packet switching**. Here, data (like the text or images of a web page) is broken into small packets, each

carrying a destination address. Each packet is sent independently across the network. As Spiceworks explains, packet switching "occupies a channel only when a packet is being moved on it," and then frees the channel for other data (Packet-Switched Network vs. Circuit-Switched Network). At the destination, these packets are reassembled in the original order (Packet-Switched Network vs. Circuit-Switched Network). Packet switching is more efficient for bursty digital data: multiple conversations can share the same links, and lost or delayed packets can be retransmitted. In summary, packet-switched networks route each packet dynamically, whereas circuit-switched networks reserve an entire path from start to end (Packet-Switched Network vs. Circuit-Switched Network) (Packet-Switched Network vs. Circuit-Switched Network).

Summary:

- Circuit switching creates a **dedicated path** (like a telephone circuit) for the entire session (Packet-Switched Network vs. Circuit-Switched Network) (Packet-Switched Network vs. Circuit-Switched Network). Data flows continuously once connected.

- Packet switching **breaks data into packets**; each packet is routed independently, and the link is used only while the packet is in transit (Packet-Switched Network vs. Circuit-Switched Network) (Packet-Switched Network vs. Circuit-Switched Network).

- Packets include address headers and are reassembled at the destination (Packet-Switched Network vs. Circuit-Switched Network), allowing efficient use of network resources.

- Circuit switching guarantees steady bandwidth (good for voice) but can **waste capacity** when idle (Packet-Switched Network vs. Circuit-Switched Network) (Packet-Switched Network vs. Circuit-Switched Network), whereas packet switching handles intermittent, bursty traffic more flexibly.

TCP/IP Protocol Layers

The Internet is built on a layered protocol model (often called the TCP/IP model). These layers (Link, Internet, Transport, Application) organize network functions (www.ietf.org). Data moves down the layers on the sender side and up the layers on the receiver side (encapsulation/decapsulation). Each layer adds its own header information. For example, **TCP/IP RFC 1122** defines the *link* (network access), *internet* (IP), *transport*, and (with RFC 1123) *application* layers (www.ietf.org).

- **Link Layer (Network Access):** Handles local delivery on the physical medium (e.g. Ethernet, Wi-Fi). The link layer wraps packets into *frames*. It adds a frame header and footer (for example, Ethernet uses a header with source/destination MAC addresses and a CRC checksum) (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). Devices like switches operate here.
- **Internet Layer (IP):** The Internet Protocol (IP) works at Layer 3. IP takes each transport-layer segment and encapsulates it in an *IP packet*. This adds source and destination IP addresses, packet length, Time-To-Live (TTL), and other fields (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). IP is *connectionless* and does not guarantee delivery, but it routes each packet toward its destination across networks.
- **Transport Layer (TCP/UDP):** This layer provides end-to-end data transport between applications. TCP (Transmission Control Protocol) offers reliable, in-order delivery with flow and congestion control. It divides application data into *segments*, adds a TCP header with source/destination ports, sequence numbers, and a checksum (Data Encapsulation and the TCP/IP Protocol Stack (System Administration

Guide: IP Services)). TCP also performs a three-way handshake (SYN, SYN-ACK, ACK) to establish a connection (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)), and retransmits segments if errors occur. UDP (User Datagram Protocol) is simpler: it adds a smaller header with ports and checksum but provides a best-effort service (no handshake, no retransmission). In short, TCP provides reliability and ordering, while UDP is faster but connectionless (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)) (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)).

- **Application Layer:** This top layer includes protocols that users see directly, like HTTP (web), DNS (names), SMTP (email), FTP (file transfer), and many others. These protocols define the format of the actual data (e.g. HTML for web pages) and run on top of TCP or UDP. For example, HTTP runs over TCP port 80 or 443, and DNS typically uses UDP port 53 (or TCP for large responses). Common application-layer protocols include HTTP, HTTPS, FTP, DNS, SMTP, and more (What is TCP/IP and How Does it Work? | TechTarget).

These layers work together. For instance, when sending an email, the message is handed to TCP, which fragments it and adds TCP headers (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). TCP then passes each segment to IP, which encapsulates it in an IP packet with IP headers (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). Finally, the link layer turns the packet into an Ethernet frame on the local network (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). At each router hop, the link-layer header is stripped and rebuilt on the next link, while IP routers read the IP header to forward the packet toward its destination.

Summary:

- The TCP/IP model has four layers: Link (network access), Internet (IP), Transport, and Application (www.ietf.org).
- Data is *encapsulated* at each layer: Transport (TCP/UDP) adds ports and checksums (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)); IP adds addressing and routing info (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)); Link layer adds frame headers and CRC checks (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)).
- TCP (Layer 4) ensures reliable delivery with sequencing and checksums (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)), while UDP provides a lightweight, connectionless service.
- Application-layer protocols (HTTP, DNS, etc.) run on top of TCP/UDP and define the actual content and services (What is TCP/IP and How Does it Work? | TechTarget)

Routers, Switches, and Hubs

In a local network or on the Internet backbone, special devices forward data: **hubs**, **switches**, and **routers**. A **hub** is a basic Layer-1 repeater: it simply rebroadcasts any incoming electrical signal to all other ports. This means every connected device sees all the traffic (leading to collisions in old Ethernet hubs) (Difference between Hub, Switch and Router | GeeksforGeeks). **Switches** are more intelligent Layer-2 devices. A switch learns which MAC addresses are on which ports and forwards each frame only to the correct destination port (Difference between Hub, Switch and Router | GeeksforGeeks) (Difference between Hub, Switch and Router | GeeksforGeeks). This greatly reduces

unnecessary traffic. Switches operate at the Data Link layer (OSI Layer 2). **Routers** operate at the Network layer (OSI Layer 3). A router reads the IP header of each packet and uses a routing table to decide the next hop. It connects multiple networks (for example, your home LAN to your ISP). Routers use IP addresses (not MAC addresses) to forward packets (Difference between Hub, Switch and Router | GeeksforGeeks) (Difference between Hub, Switch and Router | GeeksforGeeks). In summary, hubs flood all ports (Layer 1), switches forward by MAC address (Layer 2), and routers forward by IP address (Layer 3) (Difference between Hub, Switch and Router | GeeksforGeeks). In modern networks, hubs are mostly obsolete, switches are ubiquitous for connecting LAN devices, and routers direct traffic between networks (including NAT and routing on home gateways).

Summary:

- **Hub:** A simple Layer-1 device (repeater) that broadcasts incoming signals to all ports (Difference between Hub, Switch and Router | GeeksforGeeks). Inefficient and largely obsolete.
- **Switch:** A Layer-2 device that uses MAC addresses to forward frames only to the intended port (Difference between Hub, Switch and Router | GeeksforGeeks) (Difference between Hub, Switch and Router | GeeksforGeeks). Reduces collisions and isolates traffic within the LAN.
- **Router:** A Layer-3 device that examines IP addresses and routes packets to other networks (Difference between Hub, Switch and Router | GeeksforGeeks) (Difference between Hub, Switch and Router | GeeksforGeeks). Routers connect different IP networks (e.g. your LAN to the Internet) and maintain routing tables.
- In a network, switches connect multiple devices locally, while routers connect networks and manage inter-network traffic.

Routing Algorithms and BGP

Every router maintains a **routing table** and uses routing algorithms to populate it. There are two classic interior (intra-domain) routing approaches: *distance-vector* and *link-state*. **Distance-vector** protocols (like RIP) have each router periodically share its routing table with immediate neighbors. Each router calculates distances (e.g. hop count) to all destinations via Bellman-Ford updates (Difference between Distance vector routing and Link State routing | GeeksforGeeks). A drawback is slow convergence and the "count-to-infinity" problem. **Link-state** protocols (like OSPF) work differently: each router learns the entire network topology. Routers flood *link-state advertisements* to all other routers, so every router has a full map. Each then computes shortest paths (using Dijkstra's algorithm) to all destinations (Difference between Distance vector routing and Link State routing | GeeksforGeeks). Link-state converges faster and avoids routing loops, at the cost of more CPU and memory.

For routing between large networks (autonomous systems, AS), the Internet uses **BGP (Border Gateway Protocol)**. BGP is a *path-vector* protocol: each AS advertises the IP prefixes it can reach, along with attributes like the AS-PATH length. BGP routers choose the best path using these attributes rather than simple metrics (What is BGP? | BGP routing explained | Cloudflare). For example, they typically prefer paths with higher local preference or shorter AS paths (What is BGP? | BGP routing explained | Cloudflare). As Cloudflare notes, BGP "is the protocol that makes the Internet work by enabling data routing" across autonomous systems (What is BGP? | BGP routing explained | Cloudflare). It effectively picks the "most efficient" end-to-end path for traffic, hopping between ASes as needed. (BGP runs over TCP and also handles policy controls, route filtering, etc.) In summary, interior routing protocols share routes within an AS (using distance-vector or link-state techniques), while BGP handles inter-AS routing on the global Internet (What is BGP? | BGP routing explained | Cloudflare) (What is BGP? | BGP routing explained | Cloudflare).

Summary:

- **Distance-vector (RIP)**: Routers share distances with neighbors; uses Bellman-Ford to update routes (Difference between Distance vector routing and Link State routing | GeeksforGeeks). Simple but slower to adapt.
- **Link-state (OSPF)**: Routers flood network topology, then compute shortest paths with Dijkstra's algorithm (Difference between Distance vector routing and Link State routing | GeeksforGeeks). Fast convergence and loop-free.
- **BGP (Border Gateway Protocol)**: The Internet's inter-domain routing protocol. Routers (one per AS) advertise reachability with attributes. BGP selects paths using attributes (AS-path length, local pref, etc.), not just hop count (What is BGP? | BGP routing explained | Cloudflare) (What is BGP? | BGP routing explained | Cloudflare).
- BGP is vital for global routing: it "looks at all available paths" and picks the best route between autonomous systems (What is BGP? | BGP routing explained | Cloudflare).

DNS (Domain Name System) Lookup

When you enter a URL (e.g. *www.example.com*), your computer needs the server's IP address. The **DNS (Domain Name System)** is a distributed lookup service that translates domain names into IPs. DNS is often called the Internet's "phonebook" (How does DNS lookup work? | DigiCert FAQ). It is **hierarchical**: at the top are the root servers, below them the TLD (e.g. .com, .org) servers, and then the authoritative servers for each domain. The lookup process (a *recursive DNS query*) typically goes like this: the resolver (usually provided by your ISP or public DNS) first checks its cache for the domain. If not found, it queries a root server, which refers it to the correct TLD server. The resolver then queries the TLD server, which refers it to the domain's

authoritative server. Finally the authoritative server returns the IP address. As DigiCert summarizes, the resolver "will then request the IP address from root, TLD and authoritative nameservers" when it's not cached (How does DNS lookup work? | DigiCert FAQ). The IP is then returned to your client. Common DNS record types include *A* (IPv4 address), *AAAA* (IPv6 address), *CNAME* (alias), *MX* (mail server), *NS* (nameserver), etc (What Is DNS Lookup and How Does It Work). These records are what DNS servers look up and return. DNS lookups usually use UDP port 53. To improve performance, answers are cached at each stage, so repeated lookups are faster.

Summary:

- DNS maps human-readable domain names to IP addresses via a distributed hierarchy (How does DNS lookup work? | DigiCert FAQ).
- A recursive resolver checks its cache; if missing, it queries root → TLD → authoritative servers in sequence (How does DNS lookup work? | DigiCert FAQ). Each step provides referrals until the IP is found.
- Common DNS records: "A" for IPv4, "AAAA" for IPv6, "CNAME" for aliases, "MX" for mail, etc. (What Is DNS Lookup and How Does It Work).
- Caching (by ISPs, operating systems, and browsers) is used at each level to speed up lookups and reduce traffic.

Physical Layer (Fiber Optics, Submarine Cables, Data Centers)

The **physical infrastructure** of the Internet is a vast network of cables, fibers, and data centers. Crucially, more than 95% of intercontinental data travels via *submarine fiber-optic cables* on the ocean floor (Nearly all data that moves around the world goes through these undersea cables - Fast Company). These submarine cables (laid by special ships) contain many thin glass fibers. Data is transmitted as pulses of light through these fibers, allowing *terabits per second* of capacity (Nearly all data that moves around the world goes through these undersea cables - Fast Company) (Nearly all data that moves around the world goes through these undersea cables - Fast Company). For example, a single modern undersea cable can carry millions of simultaneous HD streams without slowing down (Nearly all data that moves around the world goes through these undersea cables - Fast Company). These cables form the backbone of the global Internet, connecting continents and going through strategic chokepoints (such as the Atlantic, Pacific, and around the Suez Canal). On land, data centers and terrestrial fiber networks carry traffic within and between countries. **Data centers** are facilities that house large numbers of servers, storage, and networking gear. They often contain high-speed switches/routers and redundant power/cooling. As one expert notes, data centers are essentially "the building blocks of the internet" where many networks interconnect (Data Centers Explained: The Backbone of our Digital World). Major content and cloud providers distribute data centers worldwide and connect them with high-speed fiber links (often at Internet Exchange Points) to handle user traffic. Together, the fiber-optic cables and data centers provide the physical highways and hubs for Internet data.

Summary:

- The Internet's backbone is mostly **fiber-optic cables**, especially undersea cables. These carry >95% of transoceanic data traffic (Nearly all data that moves around the world goes through these undersea cables - Fast Company).
- Submarine cables bundle dozens of glass fibers that use light to transmit data over long distances with very high bandwidth (Nearly all data that moves around the world goes through these undersea cables - Fast Company) (Nearly all data that moves around the world goes through these undersea cables - Fast Company). A single cable can carry terabits per second.
- **Data centers** are large facilities housing servers and network equipment. They interconnect networks ("building blocks of the internet" (Data Centers Explained: The Backbone of our Digital World)) and typically connect via high-speed fiber to the rest of the network.

Example Packet Journey (Typing URL to Loading Webpage)

Let's walk through a typical example: you type `http://www.example.com` in your browser and press Enter. The browser first needs the IP address of *www.example.com*. This triggers a DNS lookup (as described above) to get the IP (say `93.184.216.34`). Once the IP is known, the browser initiates an HTTP request. Here's the packet journey:

1. **TCP Handshake:** The client's OS creates a TCP segment with the HTTP request data. First, a SYN packet (with source port, dest. port 80, initial sequence number, etc.) is sent to start the connection. TCP breaks the data into segments and attaches its header (with sequence number and checksum) (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). IP then encapsulates this segment in an IP packet (adding the client's IP and the server's IP, and TTL) (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). The link layer (Ethernet) wraps the packet into a frame with MAC addresses and a CRC (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). The frame goes from your computer's NIC to your local router or switch. At the router, the Ethernet header is stripped, the IP header is examined, and the packet is forwarded toward the ISP. Each router hop repeats this: decapsulate link layer, check IP, re-encapsulate and forward on the next link. Eventually, the packet reaches the server's LAN and is delivered to the web server. The server replies with a SYN-ACK, and your client sends ACK — completing the TCP three-way handshake (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)).
2. **HTTP Request:** Once the TCP connection is established, the browser sends the actual HTTP GET request. The HTTP data is encapsulated in TCP segments (which are acknowledged by the server). The request

packet traverses routers as before. When it arrives, the server processes it and generates an HTTP response (HTML page).

3. **HTTP Response:** The response data is sent from server to client. It travels as TCP segments inside IP packets, hop by hop, back across the network. Your client reassembles the segments, checks TCP checksums (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)) and sequence numbers, and delivers the data to the browser.
4. **Rendering:** The browser now has the HTML. It may find embedded images, CSS, or scripts and repeat requests (possibly on multiple parallel TCP connections). Each of those requests follows a similar journey (DNS lookup if needed, TCP handshake if a new connection, etc.).

Throughout this journey, each packet is checked at every layer. For example, the Ethernet frame has a CRC that is verified on receipt (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)), and the TCP header has its own checksum (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). If a packet is damaged, TCP will notice a bad checksum or missing ACK and retransmit. By the end of this process, the webpage content is loaded and rendered in your browser.

Summary:

- The browser's request is sent as a TCP segment inside an IP packet, inside an Ethernet frame (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)) (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)).
- Each router forwards the packet by IP (decoding and re-encapsulating at each hop) (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)).
- A TCP three-way handshake (SYN, SYN-ACK, ACK) establishes the connection before HTTP data is sent (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)).
- The server's response follows the reverse path. TCP sequencing and acknowledgments ensure packets arrive intact and in order (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)).

Error Handling, Congestion Control, and Data Integrity

The network includes many error checks and control mechanisms to ensure reliable delivery and avoid overload. At the data-link layer, each Ethernet frame carries a cyclic redundancy check (CRC) in its trailer (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). The receiver computes the CRC on arrival; if it doesn't match, the frame is discarded. At the transport layer, TCP includes a checksum in every segment (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). Both sender and receiver use this checksum to detect corruption. If TCP notices a missing or corrupted packet (via a missing ACK), it will retransmit the data, ensuring *reliability* (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). IP also performs a header checksum, and TTL field prevents packets from looping indefinitely.

For **congestion control**, TCP uses algorithms to avoid flooding the network. For example, TCP slow-start begins by sending a small amount of data and gradually increasing it until the network capacity is found (TCP slow start - MDN Web Docs Glossary: Definitions of Web-related terms | MDN). When packet loss occurs (a sign of congestion), TCP reduces its sending rate (multiplicative decrease). In this way (slow start and congestion avoidance), TCP dynamically adapts its rate so as not to overwhelm routers (TCP slow start - MDN Web Docs Glossary: Definitions of Web-related terms | MDN) (TCP slow start - MDN Web Docs Glossary: Definitions of Web-related terms | MDN). This flow control combined with acknowledgments keeps data flowing smoothly over busy links.

Summary:

- Each layer has error checks: Link-layer CRC (frame integrity) (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)) and TCP checksum (segment integrity) (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)).
- TCP is **reliable**: it uses sequence numbers, ACKs, and retransmissions to recover lost or corrupted packets (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)).
- To prevent congestion, TCP employs algorithms (slow start, congestion avoidance) that probe and adjust the sending rate based on network feedback (TCP slow start - MDN Web Docs Glossary: Definitions of Web-related terms | MDN) (TCP slow start - MDN Web Docs Glossary: Definitions of Web-related terms | MDN).
- These mechanisms ensure data integrity and adapt to network conditions so that packets are delivered accurately despite errors or heavy traffic.

Security in Data Transmission (TLS/SSL Basics)

To protect data from eavesdropping and tampering, many Internet applications use **TLS/SSL** encryption. TLS (Transport Layer Security) is a cryptographic protocol that runs on top of TCP. When you access `https://example.com`, the client and server perform a TLS handshake before any HTTP data is sent. In this handshake, they agree on a TLS version and cipher suite, exchange and verify certificates, and use asymmetric cryptography to generate a shared secret key (An overview of the SSL/TLS handshake). Once the shared key is established, all subsequent data (e.g. the HTTP request and response) is encrypted with symmetric encryption. In short, TLS provides end-to-end security: **confidentiality** (encryption hides the data) and **integrity** (messages are authenticated to prevent modification). As the Internet Society puts it, TLS "encrypts data sent over the Internet to ensure that eavesdroppers and hackers are unable to see what you transmit" (What is TLS & How Does it Work? - Internet Society). The browser indicates a successful TLS connection with a padlock icon. In practice, TLS is used not only for secure web (HTTPS) but also for email (SMTPS/IMAPS), DNS-over-TLS, and many other applications (What is TLS & How Does it Work? - Internet Society).

Summary:

- **TLS/SSL** encrypts and authenticates data between client and server, protecting privacy (What is TLS & How Does it Work? - Internet Society).
- A handshake protocol establishes shared encryption keys: client and server agree on parameters, exchange certificates, and derive a symmetric key (An overview of the SSL/TLS handshake).
- After the handshake, all transmitted data is encrypted and integrity-protected. This is seen in browsers as HTTPS with a padlock (What is TLS & How Does it Work? - Internet Society).
- TLS uses asymmetric crypto (public keys) to set up the connection and symmetric crypto for bulk data; it secures web browsing, email, DNS, etc., against eavesdropping and tampering.

References

- R. Braden (ed.), **Requirements for Internet Hosts – Communication Layers** (RFC 1122, 1989). (Defines TCP/IP layers: link, IP, transport, application.) (www.ietf.org).
- BasuMallick, C. *"Packet-Switched Network vs. Circuit-Switched Network."* Spiceworks (2022). (Explains packet vs. circuit switching.) (Packet-Switched Network vs. Circuit-Switched Network) (Packet-Switched Network vs. Circuit-Switched Network).
- Oracle Corporation, *"TCP/IP Protocol Stack (System Administration Guide: IP Services)."* (2007) (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)) (Data Encapsulation and the TCP/IP Protocol Stack (System Administration Guide: IP Services)). (Details TCP segmentation, checksums, IP datagrams, framing, CRC, and TCP handshake.)
- Cloudflare Learning Center, *"What is BGP? (BGP routing explained)."* (Explains BGP's role and path selection.) (What is BGP? | BGP routing explained | Cloudflare) (What is BGP? | BGP routing explained | Cloudflare).
- Cormac Reid (Fast Company), *"Nearly all data that moves around the world goes through these undersea cables,"* Apr 2024. (Discusses submarine fiber-optic cables and global traffic.) (Nearly all data that moves around the world goes through these undersea cables - Fast Company) (Nearly all data that moves around the world goes through these undersea cables - Fast Company).
- Himadri Sharma, *"What Is DNS Lookup and How Does DNS Lookup Work,"* Mailmodo (2025). (DNS query process and record types.) (How does DNS lookup work? | DigiCert FAQ) (What Is DNS Lookup and How Does It Work).
- DataCenter-Jedi.com, *"Data Centers Explained: The Backbone of our Digital World."* (Describes role of data centers in the Internet.) (Data Centers Explained: The Backbone of our Digital World).

- Internet Society, *"TLS Basics."* (Describes TLS purpose and use in secure communication.) (What is TLS & How Does it Work? - Internet Society) (What is TLS & How Does it Work? - Internet Society).
- IBM Documentation, *"An overview of the SSL/TLS handshake."* (Summarizes TLS handshake steps and goals.) (An overview of the SSL/TLS handshake).
- GeeksforGeeks, *"Difference between Distance vector routing and Link State routing."* (Routing algorithm comparison.) (Difference between Distance vector routing and Link State routing | GeeksforGeeks) (Difference between Distance vector routing and Link State routing | GeeksforGeeks).
- GeeksforGeeks, *"Difference between Hub, Switch and Router."* (Device roles and OSI layers.) (Difference between Hub, Switch and Router | GeeksforGeeks) (Difference between Hub, Switch and Router | GeeksforGeeks).

Each of the above sources provides authoritative details on the Internet technologies discussed, including RFC specifications, technical manuals, and networking references.